Informe Código Dañino CCN-CERT ID-20/17

Código dañino.Petya/NotPetya

Julio de 2017

CENTRO CRIPTOLOGICO NACIONAL
2.5.4.13=Qualified Certificate: AAPP-SEP-M-SW-KPSC,
ou=sello electrónico, serialNumber=S2800155J,
o=CENTRO CRIPTOLOGICO NACIONAL, cn=CENTRO
CRIPTOLOGICO NACIONAL, c=ES
2017.07.04 12:53:22 +02'00'

LIMITACIÓN DE RESPONSABILIDAD

El presente documento se proporciona de acuerdo con los términos en él recogidos, rechazando expresamente cualquier tipo de garantía implícita que se pueda encontrar relacionada. En ningún caso, el Centro Criptológico Nacional puede ser considerado responsable del daño directo, indirecto, fortuito o extraordinario derivado de la utilización de la información y software que se indican incluso cuando se advierta de tal posibilidad.

AVISO LEGAL

ÍNDICE

1. SOBRE CCN-CERT

El CCN-CERT (www.ccn-cert.cni.es) es la Capacidad de Respuesta a Incidentes de Seguridad de la Información del Centro Criptológico Nacional, CCN (www.ccn.cni.es). Este servicio se creó en el año 2006 como el **CERT Gubernamental/Nacional** español y sus funciones quedan recogidas en la Ley 11/2002 reguladora del Centro Nacional de Inteligencia, el RD 421/2004 regulador del CCN y en el RD 3/2010, de 8 de enero, regulador del Esquema Nacional de Seguridad (ENS), modificado por el RD 951/2015 de 23 de octubre.

De acuerdo a todas ellas, es competencia del CCN-CERT la gestión de ciberincidentes que afecten a **sistemas del Sector Público**, a **empresas y organizaciones de interés estratégico** para el país y a cualquier sistema clasificado. Su misión, por tanto, es contribuir a la mejora de la ciberseguridad española, siendo el centro de alerta y respuesta nacional que coopere y ayude a responder de forma rápida y eficiente a los ciberataques y a afrontar de forma activa las ciberamenazas.

2. RESUMEN EJECUTIVO

El presente documento recoge el análisis preliminar de una campaña masiva a nivel global en varios países con distintas muestras de Ransomware de la familia **Petya/NotPetya**, con el objetivo de realizar un cifrado del sector de arranque al igual que de determinados archivos en el sistema comprometido y solicitar un rescate para recuperarlos.

Esta variante de ransomware incorpora código para realizar la explotación de la vulnerabilidad publicada por Microsoft el día **14 de marzo** descrita en el boletín MS17-010 y conocida como **ETERNALBLUE** y **ETERNALROMANCE**.

El código dañino escanea la red local del sistema comprometido en busca de nuevas máquinas a las que afectar mediante diversos métodos conectando al puerto 445 (SMB) o al puerto 139, lo que le confiere a la muestra funcionabilidad similar a la de un gusano que a diferencia de la amenaza WannaCry sólo escanea la red local y no la externa. El movimiento lateral dentro de la red utiliza una variante del payload **DOUBLEPULSAR** similar a la utilizada en la amenaza WannaCry.

A diferencia de la otra amenaza enumera la red interna mediante DHCP, obteniendo direcciones IP en el rango de la subred mediante funciones de Windows, e intenta conectarse a determinados recursos compartidos en todas las máquinas a las que pueda acceder.

Algunas variantes del código dañino van acompañadas de una herramienta llamada "Mimikatz", la cual es utilizada para acceder a la memoria del proceso que contiene las claves de los usuarios en el sistema, obteniendo de esta forma las claves de los usuarios para intentar conectar a las máquinas en la subnet.

El código dañino es una variante reescrita del código dañino "Ransom.Petya" altamente modificada para ser considerada una evolución considerable en la familia de dicho ransomware.

El código dañino, pese a contar con características de ransomware, posee capacidades destructivas ya que los sectores de arranque cifrados no pueden ser restaurados, incluso si se cumple una condición concreta los sectores del arranque son borrados en su totalidad impidiendo que el sistema operativo arranque. Del mismo modo al haber cifrado archivos en el disco éstos no pueden ser recuperados excepto desde copias de seguridad previas.

3. INFORMACIÓN DEL CÓDIGO DAÑINO

El código dañino posee un número indeterminado de versiones distintas.

En el presente informe el análisis se centrará sobre la muestra con el *hash*:

71b6a493388e7d0b40c83ce903bc6b04

4. CARACTERÍSTICAS DEL CÓDIGO DAÑINO

El código dañino examinado posee las siguientes características:

- Carga el código dañino en el sistema.
- Cifra el sector de arranque y la tabla de archivos del disco duro (MFT).
- Cifra todos los archivos en todas las unidades que cumplan un patrón de extensión.
- Se propaga mediante el exploit MS17-010 por la red local.
- Enumera la red local mediante distintos métodos buscando nuevas máquinas para infectar aunque tengan el parche MS17-010 aplicado.
- Muestra información acerca del secuestro de los archivos y solicita un rescate para su recuperación.
- Ejecuta determinados programas incluidos en el código dañino.
- Elimina los registros de logs del sistema comprometido.
- Algunas muestras ejecutan un programa para obtener las claves y credenciales de los usuarios del sistema o del dominio.
- Reinicia el equipo de varias formas distintas para cifrar su tabla de archivos.
- Posee un certificado caducado pero válido firmado por Microsoft.

5. DETALLES GENERALES

El binario tiene formato PE (*Portable Executable*), es decir, es un ejecutable para sistemas operativos Windows, concretamente para 32 bits de tipo librería dinámica ("DLL").

En la muestra analizada se ha podido observar que la fecha interna de creación del programa data del 18 de junio de 2017, aunque esta fecha podría no ser real ya que puede ser modificada.

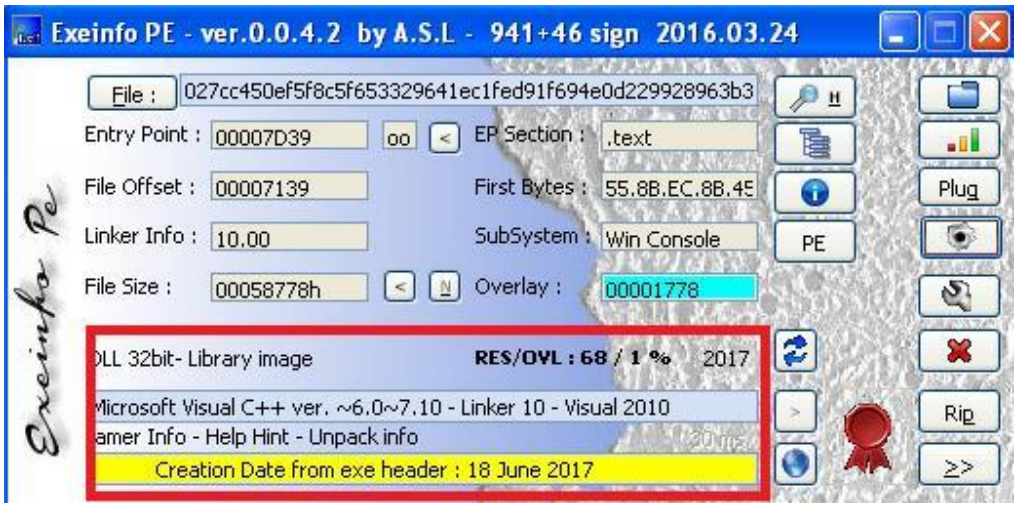


Imagen 1. Información del código dañino

6. PROCEDIMIENTO DE INFECCIÓN

La infección en el equipo se produce mediante otra máquina infectada utilizando el exploit MS17-010 y mediante la conexión al puerto 445 o puerto 139 de una máquina en la red local de la que se posean las credenciales de acceso.

Una vez ejecutado el código dañino se realizan las siguientes acciones en el equipo de la víctima:

- Extrae de sus recursos determinados componentes para la obtención de las contraseñas de usuarios en el sistema comprometido.
- Crea numerosos hilos para distintas tareas.
- Cifra el sector de arranque y reinicia el sistema para cifrar la tabla de archivos (MFT).
- Cifra todos los archivos encontrados que cumplan un patrón de extensión en todas las unidades que encuentre en el sistema comprometido.
- Procede a infectar nuevas máquinas mediante el exploit MS17-010 que no hayan sido parcheadas, y a aquellas que consiga detectar a las que pueda conectar mediante el puerto 445 o 139 y tenga sus credenciales.
- Se borra del sistema comprometido de forma segura para no dejar rastro de su existencia en un análisis forense de la máquina comprometida.
- Borra todos los logs del sistema comprometido.

7. CARACTERÍSTICAS TÉCNICAS

El código dañino es una librería dinámica ("DLL") cuya actividad comienza mediante una función exportada llamada "perfc" y con el ordinal número 1.

La primera acción que realiza es obtener una serie de privilegios en el sistema comprometido ajustando su propio token del proceso mediante la función "AdjustTokenPrivileges".

Los privilegios que intenta obtener son los siguientes:

SeShutdownPrivilege
SeDebugPrivilege
SeTcbPrivilege

El primer privilegio de la tabla lo necesita para poder apagar o reiniciar el sistema posteriormente, el segundo privilegio para poder acceder a cualquier proceso del sistema como al disco en modo físico y el último para poder interactuar como si fuera el propio sistema operativo.

El primer privilegio lo obtendrá aunque no se sea usuario de tipo administrador, sin embargo el segundo privilegio requiere que el usuario que este ejecutando el

código dañino pertenezca a ese grupo de usuarios, y el último, por defecto está sólo permitido al usuario "SYSTEM".

El código dañino se replica mediante el exploit ETERNALBLUE por lo que se podría ejecutar como SYSTEM, obteniendo todos los privilegios.

El código dañino guarda un registro en una variable de la cantidad de privilegios que se haya podido otorgar.

```
NewPetyaEnablePrivilegesAndGetModuleFileName proc near
                                        ; CODE XREF: perfc_1+10↓p
                push    esi
                xor     esi, esi
                cmp     _flag_is_started, esi
                jnz     short _exit
                call    ds:GetTickCount
                push    offset aSeshutdownpriv ; "SeShutdownPrivilege"
                mov     NewPetyaGetTickCountVar, eax
                call    NewPetyaGetCurrentTokenAndAdjustPrivilege
                test    eax, eax
                jz      short _enable_debug_privilege
                inc     esi             ; 1 privilege

_enable_debug_privilege:                ; CODE XREF: NewPetyaEnablePrivilegesAndGetModuleFileName+22↑j
                push    offset aSedebugprivile ; "SeDebugPrivilege"
                call    NewPetyaGetCurrentTokenAndAdjustPrivilege
                test    eax, eax
                jz      short _enable_tcb_privilege
                or      esi, 2          ; 2 privileges (shutdown and debug)

_enable_tcb_privilege:                  ; CODE XREF: NewPetyaEnablePrivilegesAndGetModuleFileName+31↑j
                push    offset aSetcbprivilege ; "SeTcbPrivilege"
                call    NewPetyaGetCurrentTokenAndAdjustPrivilege
                test    eax, eax
                jz      short _prepare_to_get_processes
                or      esi, 4          ; esi or with 4 (all privileges already)

_prepare_to_get_processes:              ; CODE XREF: NewPetyaEnablePrivilegesAndGetModuleFileName+42↑j
                mov     NewPetyaPrivilegesThatCanGetValue, esi
```

Ilustración 2. Obtención de privilegios necesarios

Posteriormente el código dañino enumera los procesos del sistema y por cada uno de ellos calcula un hash con un algoritmo propietario. En el caso de que uno de los procesos se llame "avp.exe" (del antivirus AVP) se pondrá un variable a un valor determinado. Esta variable será comprobada posteriormente y, si se ha cumplido la condición, en lugar de cifrar el MBR y la MFT procederá a borrar los sectores rellenándolos de basura.

A continuación el código dañino lee en su totalidad un buffer reservado dinámicamente, para proceder a ejecutarse en la región de memoria recién creada. Una vez ejecutándose desde la memoria reservada procede a sobreescribirse en disco con caracteres nulos y a borrarse mediante la función "DeleteFileA", realizando de esta forma un borrado seguro para evitar futuros análisis forenses del disco.

Tras ello obtiene todas las direcciones de memoria de las funciones que necesitará de determinadas librerías mediante el uso de las funciones "LoadLibraryA" y "GetProcAddress".

Una vez realizadas estas acciones el código dañino obtiene su línea de argumentos y comprueba si ha recibido algún argumento. El código dañino soporta un argumento numérico opcional, el cual será usado posteriormente para indicar el tiempo a esperar antes del reinicio forzado del sistema. En el caso de que dicho argumento no sea introducido el código dañino asumirá que dicho valor es 60.

A continuación procede a comprobar si posee tanto el privilegio de apagado como el de depuración, y en el caso de que así sea crea un archivo en el directorio de %WINDOWS% con su mismo nombre sin la extensión. Antes de la copia comprueba que ya no exista en el directorio, en el caso de que exista con el mismo nombre el código dañino finaliza su ejecución sin realizar ningún tipo de carga dañina.

Si posee ambos permisos procede a acceder a la unidad "C:" y obtiene sus dimensiones mediante el uso de la función "DeviceIoControl":

```
push    40000000
push    9A43C4                                  ASCII "\\.\C:"
call    [99D160]                                kernel32.CreateFileA
mov     edi, eax
cmp     edi, esi
je      short 00998DE6
push    esi
lea     eax, [esp+10]
push    eax
push    18
lea     eax, [esp+1C]
push    eax
push    esi
push    esi
push    70000                                   ASCII "Actx "
push    edi
call    [99D1AC]                                kernel32.DeviceIoControl
```

Ilustración 3. Obtención de los sectores del disco de la unidad C:

A continuación genera una clave de forma aleatoria mediante las funciones seguras de la librería "crypt32.dll". No se conocen vulnerabilidades para la función de "CryptGenRandom". La clave generada es de 60 bytes. Con dicha clave genera un identificador único de la víctima.

Posteriormente lee el primer sector del disco duro accedido de forma física a un buffer de memoria reservado con el tamaño del sector, tras la lectura del sector cifra todos los bytes del sector en memoria con el valor "7" (no confundir con el número 7).

A continuación genera otro valor aleatorio de 0x20 bytes usando la función segura de "CryptGenRandom". Esta clave será la utilizada para cifrar con el algoritmo "Salsa20" la tabla de archivos (MFT) en el siguiente reinicio.

Posteriormente escribe en la superficie del disco el código del arranque de la carga dañina y todo su "payload".

A su vez escribe en el sector 34 el código de arranque del primer sector original cifrado con el valor "7".

Tras haber realizado la escritura en el disco o en el caso de que no tuviera los privilegios adecuados continua con el cálculo de la hora a la que realizará el reinicio forzado del sistema. Para ello utiliza la función "GetLocalTime" y el valor que se le introdujo como argumento al llamar a la función exportada (en el caso de que no se le pasase valor se asigna el valor por defecto de 60). Obtiene la versión del sistema operativo mediante "GetVersionEx", si ésta es Windows XP o inferior utiliza el comando "shutdown.exe /r /f" y la hora adecuada añadiendo a la actual el valor indicado como argumento.

En el caso de que sea superior a XP se comprueba si tiene el privilegio de "SeTcbPrivilege", si es así prepara una tarea programada como el usuario "SYSTEM", en caso contrario con el usuario actual mediante el comando "schtasks.exe" con la hora indicada por la suma de la hora local más el valor del argumento.

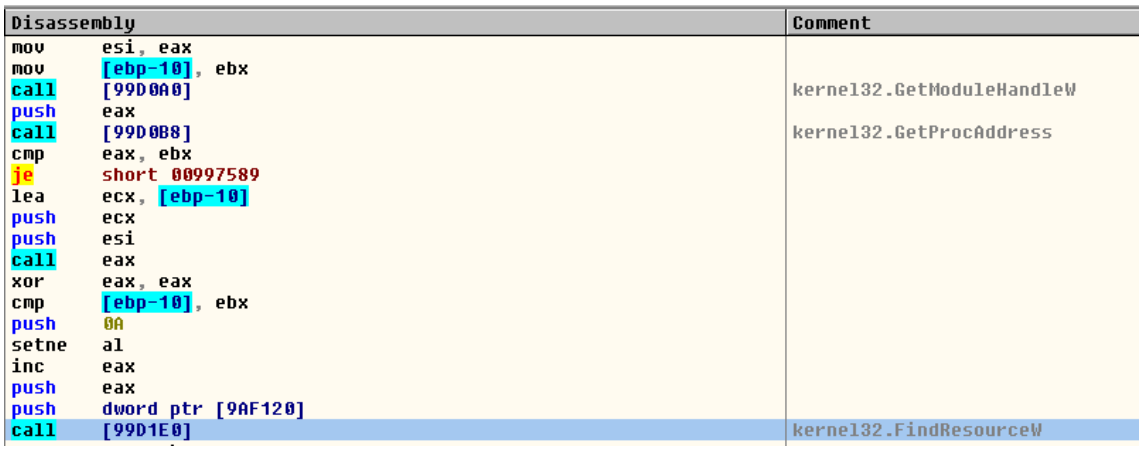
Aunque posteriormente intentará realizar un reinicio forzado utilizando diversos métodos, incorpora este comando para asegurarse que el sistema se reiniciará.

Para ejecutar el comando obtiene el intérprete de comandos del sistema comprometido mediante la función "GetEnvironmentVariableW" o si la función falla asumirá que el intérprete es el programa "cmd.exe". Obtiene el directorio de sistema y crea la cadena para llamar al intérprete, pasándole como argumento el programa para realizar el reinicio temporizado.

A continuación realiza un "Sleep" que, por un fallo de su programación, siempre es de 0 milisegundos.

Posteriormente el código dañino crea un hilo para comenzar a enumerar la red local en busca de nuevas máquinas a las que comprometer. Este hilo se explica con más detalle en el apartado 10.CONEXIONES DE RED del presente informe.

Tras la creación del hilo el código dañino comprueba que tenga el privilegio de depuración y si lo posee extrae un nuevo binario comprimido desde sus recursos. Para ello comprueba si se está ejecutando en un entorno de 64bits o no, y dependiendo del caso extrae el recurso llamado "1" (en el caso de 32bits) o el recurso llamado "2" (en el caso de 64bits).



Disassembly		Comment
mov	esi, eax	
mov	[ebp-10], ebx	
call	[99D0A0]	kernel32.GetModuleHandleW
push	eax	
call	[99D0B8]	kernel32.GetProcAddress
cmp	eax, ebx	
je	short 00997589	
lea	ecx, [ebp-10]	
push	ecx	
push	esi	
call	eax	
xor	eax, eax	
cmp	[ebp-10], ebx	
push	0A	
setne	al	
inc	eax	
push	eax	
push	dword ptr [9AF120]	
call	[99D1E0]	kernel32.FindResourceW

Ilustración 4. Acceso al recurso embebido "1" ó "2"

Una vez accedido el recurso, se descomprime en memoria, y se crea en la carpeta temporal con un nombre aleatorio y con el atributo de oculto.

Después genera una cadena de texto desde un CLSID generado aleatoriamente, esta cadena de texto se utiliza para concatenarla a la cadena siguiente:

"\\.pipe\<cadena_del_CLSID>"

Posteriormente se ejecuta el archivo recién creado pasándole como argumento la cadena indicada en la tabla anterior. El proceso recién ejecutado accederá al proceso "lsass.exe" del sistema para intentar obtener todas las credenciales posibles de los usuarios activos o que hayan estado activos en el sistema ya sea de forma local o desde un sistema remoto y los trasmitirá a la librería dinámica mediante el "pipe" creado con el nombre indicado anteriormente.

Este "pipe" es creado en un hilo, el cual comprueba que no haya cambiado el estado del "pipe" cada segundo y en el caso de que haya cambiado leerá su contenido buscando la información enviada desde el nuevo proceso.

El código dañino espera mediante "WaitForSingleObject" a que el proceso haya terminado su ejecución, finalizando también el hilo dedicado a la lectura del "pipe".

Una vez finalizada su ejecución, el código dañino hace un borrado seguro del ejecutable sobreescribiendo su contenido con caracteres nulos y usando "DeleteFileW".

El código dañino borra todos los buffers de memoria en donde ubicó el ejecutable descomprimido rellenándolo con caracteres nulos, evitando así que con un volcado de memoria se pueda extraer el ejecutable recién ejecutado.

A continuación el codigo dañino extrae el recurso llamado "3" y lo copia en la carpeta de %WINDOWS% bajo el nombre de "dllhost.dat". Este ejecutable recién creado no es dañino, es el ejecutable legítimo de la utilidad llamada "PSEXEC" de Microsoft. El código dañino lo utilizará para, con las credenciales robadas, intentar conectarse a equipos remotos, copiarse en ellos y ejecutarse.

Posteriormente el código dañino crea una serie de hilos destinados a enumerar de diversas formas la red interna en donde se encuentre el sistema comprometido. Utiliza la enumeración mediante conexiones TCP, DHCP, al igual que intenta acceder al recurso "admin$" de todas las máquinas que encuentre. También accede al contenedor de credenciales para obtener todas las posibles.

Tras crear los hilos, el código dañino procede a enumerar todas las posibles letras de unidades existentes, y en todas aquellas que encuentre que representen a un disco duro creará un hilo para cifrar todos los archivos que encuentre en dicha unidad que cumplan un patrón de extensión.

Después de enumerar todas las unidades procede a borrar todos los registros del sistema ("logs") para no dejar ninguna evidencia de su actividad.

A continuación procede a esperar mediante una llamada a "Sleep" de gran duración, calculado del valor introducido como parámetro y multiplicado por 60000.

Tras esta espera intentará reiniciar el sistema de forma forzada en el caso de que haya podido otorgarse el permiso de apagar el sistema. Para ello usa, en primer lugar, la función no documentada de "NtRaiseHardError" para producir un "pantallazo azul".

En el caso de que dicha función no pueda ser obtenida o falle, usará la función "InitiateSystemShutdownExW", si ésta última falla también usará la función "ExitWindowsEx".

En el caso de que no tenga el privilegio para poder apagar el sistema, o fallen las 3 funciones anteriores, procede a finalizar su proceso.

8. CIFRADO Y OFUSCACIÓN

8.1 CIFRADO

El código dañino posee dos sistemas de cifrado: uno para el cifrado del MBR y de la MFT, y otro para cifrar los archivos que encuentre en las unidades de tipo disco duro que cumplan con un patrón de extensión.

8.1.1 CIFRADO DEL MBR Y MFT

El algoritmo utilizado en este cifrado es el mismo que usaban las versiones previas de la familia del ransomware, el algoritmo "Salsa20". Sin embargo, a diferencia de las anteriores versiones, los autores del código dañino han modificado la constante "SIGMA" del algoritmo.

La constante original es el texto "expand 32-byte k", sin embargo, en el código dañino se ha cambiado a la siguiente cadena "1nvalid s3ct—i--d". Este cambio ocasiona que se tengan que realizar pequeños cambios en cualquier implementación que quiera usar el mismo algoritmo.

A pesar de ello, el código dañino genera el identificador único para la víctima desde una llamada a la función "CryptGenRandom", le realiza un cambio, y no lo modifica. Es por ello que la recuperación del MBR y de la MFT del sistema comprometido es imposible, aunque se pague el rescate.

8.1.2 CIFRADO DE LOS ARCHIVOS

El código dañino cifra los archivos que posean cualquiera de las siguientes extensiones:

.3ds .7z .accdb .ai .asp .aspx .avhd .back .bak .c .cfg .conf .cpp .cs .ctl .dbf .disk .djvu .doc .docx .dwg .eml .fdb .gz .h .hdd .kdbx .mail .mdb .msg .nrg .ora .ost .ova .ovf .pdf .php .pmf .ppt .pptx .pst .pvi .py .pyc .rar .rtf .sln .sql .tar .vbox .vbs .vcb .vdi .vfd .vmc .vmdk .vmsd .vmx .vsdx .vsv .work .xls .xlsx .xvd .zip

El algoritmo utilizado para cifrar los archivos es AES 128-CBC. Se genera una clave aleatoria mediante el uso de las funciones criptograficas de Windows.

El código dañino cifra como máximo 1.048.576 bytes del archivo. Además no cifra los archivos que se encuentren en la carpeta de %WINDOWS%.

A diferencia de otros programas de su misma categoría no accede a los archivos mediante las funciones "ReadFile" y "WriteFile" sino que utiliza las funciones de mapeo de archivos para agilizar el proceso de cifrado.

Una vez cifrados todos los archivos exporta la clave AES utilizada en un "blob" cifrado con la clave RSA pública que lleva embebida el código dañino.

A continuación escribe una nota de rescate en un archivo llamado "README.TXT" en donde figuran las indicaciones para poder recuperar los archivos y el identificador único que representa el "blob" de la clave AES en formato texto.

De esta forma los autores del código dañino podrían descifrar el "blob" con la clave RSA privada que poseen y así poder obtener la clave AES utilizada para cifrar los archivos tras el pago del rescate solicitado.

9. PERSISTENCIA EN EL SISTEMA

El código dañino no crea ninguna entrada en el Registro de Windows ni tiene ningún metodo de persistencia en el sistema comprometido.

10. CONEXIONES DE RED

El código dañino utiliza el *exploit* MS17-010 para propagarse hacia todas las máquinas que no tengan parcheada esta vulnerabilidad.

Se utilizan tanto los exploits "EternalBlue" y "EternalRomance" como diversos métodos para poder acceder a las máquinas enumeradas de la red local.

A diferencia del código dañino "WannaCry" solo enumera la red local.

10.1 PROPAGACIÓN POR RED LOCAL

Para propagarse por la red local, aparte de los exploits anteriormente citados, utiliza distintos métodos:

- Enumeración de conexiones TCP
- Escaneo de direcciones usando el protocolo ARP
- Enumeración de máquinas usando DHCP
- Por cada máquina encontrada procede a comprobar si puede conectar al puerto 445 o al puerto 139.
- Utiliza la herramienta legitima "PSEXEC" para conectarse con credenciales robadas de la memoria del proceso "lsass.exe" a máquinas remotas.
- Utiliza la utilidad "WMIC" para crear su proceso en máquinas remotas con las credenciales robadas.
- Enumera recursos de red ya montados.
- Enumera recursos de red no montados que hayan sido utilizados anteriormente por el sistema operativo. En este caso el código dañino procede a montarlos y usarlos.
- Intenta conectarse al recurso "admin$" en cada máquina enumerada con las credenciales robadas.
- Enumera todos los adaptadores de red en el sistema comprometido para conocer toda su subred.

11. ARCHIVOS RELACIONADOS

El código dañino puede presentar una serie de archivos en el sistema comprometido dependiendo de su estado de ejecución, a continuación se listan los archivos que pueden existir:

<varía>			
Nombre	**Fecha Creación**	**Tamaño *bytes***	**Hash SHA1**
<varía>	<varia>	362360	34f917aaba5684fbe56d3c57d48ef2a1aa7cf06d
README.TXT	<varia>	<varía>	<varía>
<%TEMP%>			
Nombre	**Fecha Creación**	**Tamaño *bytes***	**Hash SHA1**
<varía.tmp>	<varía>	56320	38e2855e11e353cedf9a8a4f2f2747f1c5c07fcf
<varía.tmp>	<varia>	47616	56c03d8e43f50568741704aee482704a4f5005ad
<%WINDOWS%>			
Nombre	**Fecha Creación**	**Tamaño *bytes***	**Hash SHA1**
dllhost.dat	<varia>	381816	cd23b7c9e0edef184930bc8e0ca2264f0608bcb3
<varía>	<varia>	0	da39a3ee5e6b4b0d3255bfef95601890afd80709

12. DETECCIÓN

Para detectar si un equipo se encuentra, o ha estado infectado, se ejecutará alguna de las herramienta de Mandiant como el "Mandiant IOC Finder" o el colector generado por RedLine con los indicadores de compromiso generados para su detección.

El código dañino, por las características que posee, sólo puede ser detectado una vez ejecutó su carga dañina.

Dependiendo del caso el código dañino podrá reiniciar con éxito el sistema comprometido cifrando el MBR y la MFT. En otros casos no podrá acceder al disco duro de forma física con lo que sólo cifrará los archivos que cumplan determinadas extensiones.

Si se encuentran archivos a los cuales no se puede tener acceso y que posean alguna de las extensiones indicadas en la tabla del apartado 8.1.2 CIFRADO DE LOS ARCHIVOS del presente informe, es un posible indicador de compromiso del sistema.

La existencia de un archivo en cada unidad cifrada llamado "README.TXT" con información acerca del secuestro de los archivos es un signo de compromiso.

Del mismo modo si el equipo se reinició y se muestra una pantalla similar a la siguiente es un claro indicio de que el sistema está infectado.

En el caso de que el código dañino haya ejecutado la carga dañina del cifrador de archivos se podrá ver la siguiente pantalla en el idioma del sistema:

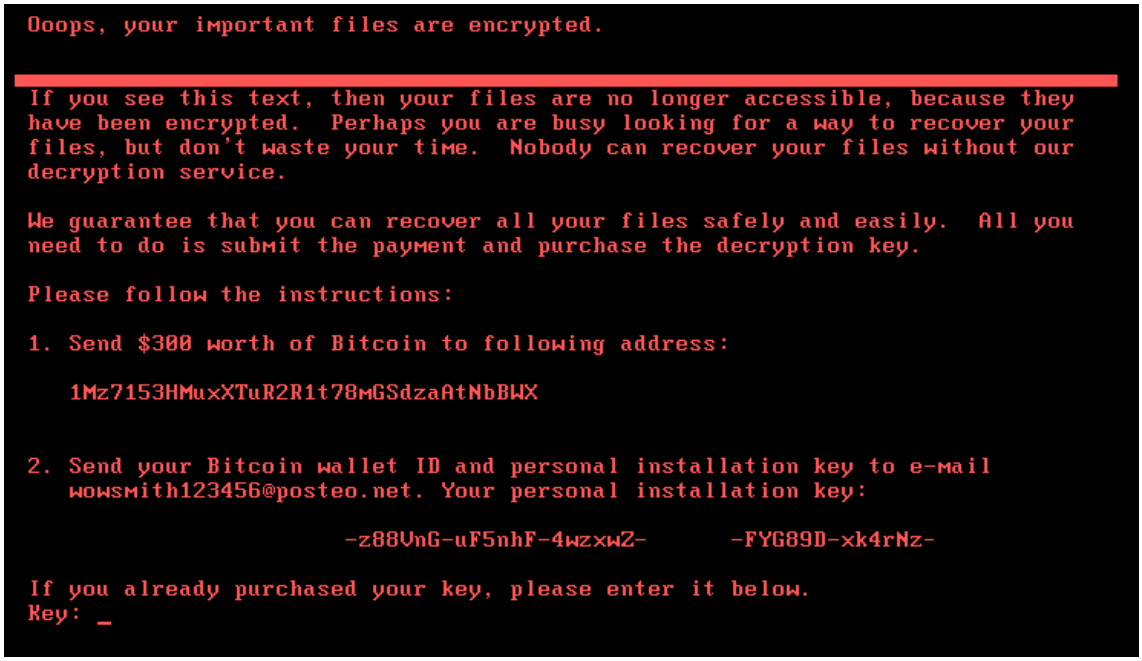


Imagen 5. Pantalla de cifrado del MBR y la MFT

13. DESINFECCIÓN

La desinfección del código dañino no es posible una vez se ha iniciado el proceso de infección.

El código dañino no escribe ninguna entrada en el registro ni crea ningún archivo que no borre posteriormente, excepto la nota de secuestro "README.TXT" en cada una de las unidades en las que haya podido cifrar archivos.

Aparte de ese archivo los únicos archivos que se conservan son:

- El archivo con el mismo nombre del código dañino ejecutado sin extensión en el directorio de %WINDOWS%.
- El archivo en el directorio de %WINDOWS% con el nombre "dllhost.dat", que es la herramienta legítima "PSEXEC".

Si el código dañino reinició el sistema y pudo cifrar los sectores de inicio del disco no se puede recuperar la información teniendo que restaurar todo el sistema operativo desde copias de seguridad o nuevas instalaciones.

Los archivos que hayan sido cifrados en las unidades tampoco pueden ser recuperados actualmente con ninguna herramienta gratuita. No se debe pagar el secuestro solicitado pues la cuenta de correo indicada en la nota ya no se encuentra operativa.

En la fecha actual no existe ningún descifrador gratuito que permita recuperar la información cifrada, por lo que se recomienda restaurar dicha información desde copias de seguridad existentes.

14. REFERENCIAS

- https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
- https://securelist.com/expetrpetyanotpetya-is-a-wiper-not-ransomware/78902/

15. INFORMACIÓN DEL ATACANTE

La única información que se posee acerca del atacante es la dirección de correo, ahora **ya no operativa**, indicada en las notas de secuestro:

wowsmith123456@posteo.net

16. REGLAS DE DETECCIÓN

16.1 YARA

```
rule Ransomware_NotPetya
{
meta:

author=”CCN-CERT”
description = “Regla para detectar la amenaza Ransomware_NotPetya”
version = “1.0”

strings:

$a1 = “1Mz7153HMuxXTuR2R1t78mGSdzaAtNbBWX” fullword ascii
$a2 = “wowsmith123456@posteo.net.” fullword wide

condition:

(uint16(0) == 0x5A4D) and (all of them)
}
```